

1/PRTS

**Method and arrangement for the power-efficient control
of processors**

5 The invention relates to a method for functionally
controlling the program and/or data flow in digital
signal processors and processors having respective
closed modules which are separate from one another, are
intended for program and data flow control and operate
10 in parallel arithmetic units.

Processors whose architecture has a slice structure are
gaining increasing importance in digital signal
processors (DSP). In this case, data paths are combined
15 to form slices, a signal processing operation in a
first slice being carried out independently of the
signal processing that is taking place in a parallel
manner in a second slice.

20 If operations are carried out in the parallel
arithmetic units of these digital signal processors
using the SIMD instruction type, the problem arises in
the prior art that the algorithms used in this case are
often not suited to the parallel signal processing in
25 all of the slices.

In the case of the signal processing in the individual
slices, for example, the results obtained can therefore
usually be provided only at different points in time or
30 after a different number of processor clock cycles in
the respective slice as a result of the respective
different algorithms used there.

The system of processing instructions in a manner that
35 concurs with the other SIMD slices either cannot be
implemented at all or can be implemented only with a
high outlay.

This necessarily high outlay occurs, on the one hand,

- 2 -

in terms of software, as additional programs which are to be executed and organize the different waiting times for the slices in order to provide the results in a parallel manner.

5

This high outlay arises, on the other hand, in the hardware, as heavy processor and memory utilization that reduces the processor performance. This reduction may be averted, for example, by expanding the memory but this signifies an increase in the outlay on hardware.

It proves to be disadvantageous in the prior art that, in order to necessarily adapt the algorithms to the SIMD instruction type during the signal processing, primarily in the slices with their associated data paths, these slices and the additional associated VLIW architecture of the processor have to be supplied, to a considerable extent, with no-operation instructions (NOP).

This not only renders the power-increasing effects of using the SIMD instruction type ineffective but also requires an additional outlay on hardware and software in order to adapt the algorithms.

The formulated object according to the invention is thus to individually adapt the signal processing (when the SIMD instruction type is used) in the individual data paths in a power-efficient manner and, in particular, to minimize the occurrence of NOP instructions with which the VLIW architecture of the processor must be supplied.

The formulated object is achieved according to the invention by means of the fact that the parallel signal processing - as a result of the SIMD instructions which are converted by the PCU - of the processor is

- 3 -

individually controlled, in a respective data path (DP) of a first and a second slice, by means of a "single slice halt" state that is output by an SSM register bank for each slice.

5

In this case, the controlling effect of the "single slice halt" state that has been output is achieved by the bits (which are assigned to the first and second slices) of the SSM register bank switching the register
10 clock supply via the respectively associated first and second gated clock cells.

As a result, the associated input register and/or accumulator and/or pipeline control register is/are
15 stopped in the meantime depending on the state of the signal processing occurring in the slice of the data path.

This functioning is enabled only by the "single slice halt" state that has been output being discontinued
20 when a further SIMD instruction is converted.

The register file unit (RFU) and the memory access register of the processor remain in operation
25 irrespective of the "single slice halt" state that has been output. The PCU can in this case write to the SSM register bank of the PCU at any time.

This solution is aimed at beginning with the individual
30 calculations in a parallel manner in the slices of the data paths of the processor, in accordance with the SIMD instruction type.

However, as a result of the different calculation
35 processes, the intermediate and/or final results in the slices are provided at different points in time in the pipeline control registers, accumulators and result registers of the associated data paths.

After the intermediate and/or final result values have been provided, a further signal processing operation that is no longer laden with results is thus prevented
5 in the data paths which are associated with the individual slices.

The signal processing is continued in a parallel manner in all of the data paths of the slices if a start is
10 made on processing a further SIMD instruction.

A supplementary embodiment of the solution (according to the invention) of the formulated object consists in controlling the clock supply for the VLIW unit, by
15 means of a software-dictated output of the state from the program flow of the processor, in such a manner that, as a result, partial instruction words which are currently present in the VLIW unit are subsequently provided in the latter for multiple use at the
20 functional units.

This solution according to the invention advantageously becomes effective if necessary adaptation of the algorithms to the SIMD instruction type during the
25 signal processing makes it necessary for the data paths and the associated VLIW architecture of the processor to be supplied with no-operation instructions (NOP) or similar instructions with a high repetition rate. In this case, avoiding the generation of identical VLIWs
30 reduces the amount of memory space used and keeps the computing load of the processor low, with the result that the computing power is efficiently available for the important calculations.

35 One advantageous variant of the supplementary embodiment of the solution according to the invention consists in interrupting the generation of further VLIWs in the VLIW unit by the PCU being informed of a

- 5 -

VLIW WAIT command via an advance signal line and this command being applied to the PCU in the next clock cycle, the PCU then switching the clock supply for the VLIW unit by means of a "VLIW WAIT" signal line and a
5 third gated clock cell.

This solution is aimed at being able to realize debugging routines in software tests by it being possible to set and start software breakpoints in the
10 program code.

The invention will be explained in more detail below with reference to an exemplary embodiment for outputting a single slice halt state. The figure of the
15 drawing contains a block diagram of the processor, in which the parts with the associated functional units which relate to the solution according to the invention are given.

20 In the event of the "single slice halt" state being output, it is a prerequisite that an SIMD instruction is output by the VLIW unit 2 via the SIMD control bus 12. This individual SIMD instruction triggers multiple data processing in the respective data path 14 of the
25 first and second slices 18; 19.

The results are provided at different points in time in the associated accumulator 8. In this case, a respective bit (which is assigned to the first and
30 second slices 18; 19) of the SSM register bank 13 is set.

The signal allocation of this bit is supplied, via the first and/or second gated clock cell 3; 4, to the data
35 path 14 (that is respectively associated with the first and second slices 18; 19) and individually controls the signal processing in the first and second slices 18; 19 in that the clock supply at the associated input

- 6 -

register and thus also the signal processing are prevented when a result is present in this slice.

When a further SIMD instruction is output on the SIMD control bus 12, for example after the last result worked out in one of the slices has been provided, the respective bit of the SSM register bank 13 is reset and all of the data paths begin the next signal processing operation by reading in the data provided by the RFU 11 at their input registers.

The signal processing in the individual slices of the data paths 14 is thus advantageously adapted to the requirements of parallel processing of the SIMD instructions.

**Method and arrangement for the power-efficient control
of processors**

List of reference symbols

- | | |
|----|---|
| 1 | Processor |
| 2 | VLIW (<u>V</u> ery <u>L</u> ong <u>I</u> nstruction <u>W</u> ord) unit |
| 3 | First gated clock cell |
| 4 | Second gated clock cell |
| 5 | AGU (<u>A</u> ddress <u>G</u> enerating <u>U</u> nit) |
| 6 | PCU (<u>P</u> rocess <u>C</u> ontrolling <u>U</u> nit) |
| 7 | Clock supply line |
| 8 | Accumulator |
| 9 | Further processing unit (with gated clock cell) |
| 10 | Register of the further processing unit |
| 11 | RFU (Register File Unit) |
| 12 | SIMD control bus |
| 13 | SSM (<u>S</u> ingle <u>S</u> lice <u>M</u> ode) register bank |
| 14 | Data path |
| 15 | SIMD data path control line |
| 16 | Advance signal line |
| 17 | VLIW WAIT signal line |
| 18 | First slice |
| 19 | Second slice |
| 20 | Third gated clock cell |

**Method and arrangement for the power-efficient control
of processors**

5

Patent Claims

1. A method for functionally controlling the program
and/or data flow in digital signal processors and
processors having respective closed modules which
10 are separate from one another, are intended for
program and data flow control and operate in
parallel arithmetic units, wherein, as a result of
the SIMD instructions which are converted by the
PCU (6), the parallel signal processing of the
15 processor (1) is individually controlled, in a
data path DP (14) that is respectively associated
with the first and second slices (18);(19), by
means of a "single slice halt" state that is
output by an SSM register bank (13), the
20 controlling effect of the "single slice halt"
state that has been output being achieved by the
bits (which are assigned to each slice) of the SSM
register bank (13) switching the register clock
supply via the respective first and second gated
25 clock cells (3);(4) and, as a result, the
functioning of the assigned input register and/or
accumulator and/or pipeline control register being
stopped in the meantime depending on the state of
the signal processing occurring in the DP (14)
30 associated with the respective slice and said
functioning being enabled again only by the
"single slice halt" state that has been output
being discontinued as a result of a further SIMD
instruction being converted,
35 and wherein the register file unit (RFU) (11) and
the memory access register of the processor (1)
remain in operation irrespective of the "single
slice halt" state that has been output and the PCU

- 9 -

(6) can in this case write to the SSM register bank (13) of the PCU at any time.

2. A method for functionally controlling the program
5 and/or data flow in digital signal processors and
processors having respective closed modules which
are separate from one another, are intended for
program and data flow control and operate in
10 parallel arithmetic units, wherein the clock
supply for the VLIW unit (2) is controlled, by
means of a software-dictated output of the state
from the program flow of the processor (1), in
such a manner that, as a result, partial
15 instruction words which are currently present in
the VLIW unit (2) are subsequently provided in the
latter for multiple use at the functional units.
3. The method as claimed in claim 2, wherein the
20 generation of further VLIWs in the VLIW unit (2)
is interrupted by the PCU (6) being informed of a
VLIW WAIT command via an advance signal line (16)
and this command being applied to the PCU (6) in
the next clock cycle, the PCU (6) then switching
25 the clock supply for the VLIW unit (2) by means of
a "VLIW WAIT" signal line (17) and a third gated
clock cell (20).

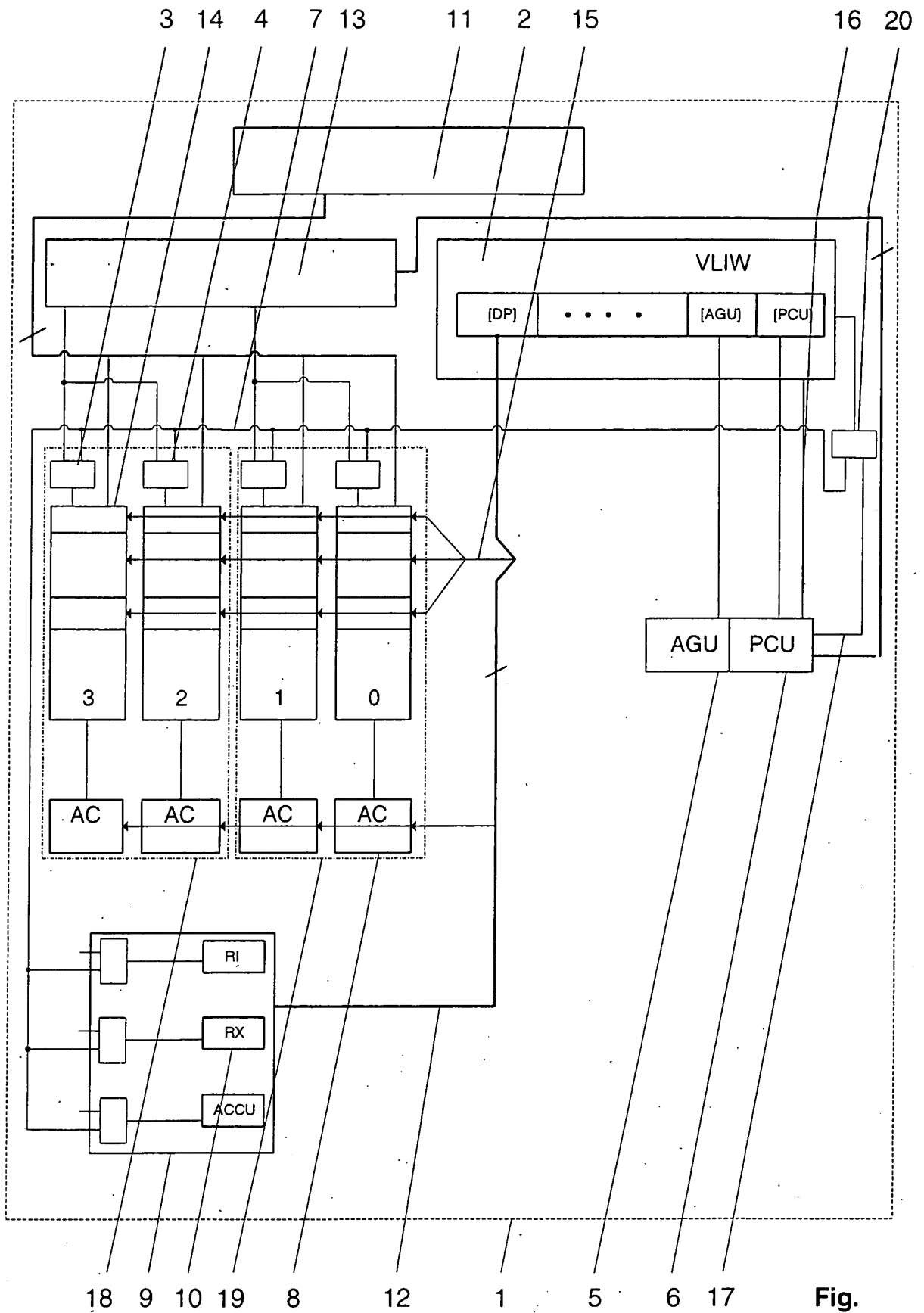


Fig.